REMARKS

IN THE SPECIFICATION

According to the foregoing, the specification is amended to correct errors. Support for the specification amendments can be found, for example, in FIGS. 10 and 14, or support is clearly found in case of correcting spelling errors.

AMENDMENTS TO THE DRAWINGS

Attached are 2 drawing replacement sheets for FIGS. 10 and 12 (total of 2 sheets).

The 2 drawing replacement sheets for FIGS. 10 and 12 are corrected drawings to correct errors, as follows.

In FIGS. 10 and 12, at operation 3, the spelling of the word "WEAL" is corrected to --WEAK--.

In FIGS. 10 and 12, at operation 7, the following phrase is corrected as indicated herein: --~~MAKE~~CHANGE THE STATE OF DATA FROM ~~MEMORY~~OTHER CACHE MEMORIES INTO S AND STORE THE DATA IN THE CACHE--. Support for the claim amendments can be found, for example, in paragraph 10 of the present substitute specification.

Accordingly, entry of the 2 drawing replacement sheets for FIGS. 10 and 12 is respectfully requested.

REJECTIONS

Claims 1-16 are pending, of which claims 1, 4, 7, 14, 15, and 16 are independent.

Claims 7-10 and 16 are maintained as rejected under 35 USC 112, first paragraph, for failing to comply with the enablement requirement.

Further, independent claims 4 and 15 are rejected under 35 USC 112, first paragraph, for failing to comply with the enablement requirement.

Claim 12 is rejected under 35 USC 112, second paragraph, as being indefinite.

Claims 1-3 and 14 are rejected under 35 USC 103(a) as being unpatentable over Arimilli (US Patent No. 6,374,330) in view of Bourekas (US Patent No. 6,128,703). Arimilli and Bourekas are newly cited, and, thus, newly relied upon to reject independent claims 1 and 14.

Dependent claims 5 and 6 are rejected under 35 USC 103(a) as being unpatentable over Arimilli, Bourekas, and further in view of Prudvi (US Patent No. 6,378,048). Prudvi is newly cited, and thus, newly relied upon.

Dependent claims 11 and 12 are rejected under 35 USC 103(a) as being unpatentable over Arimilli, Bourekas, and in view of Gornish (US Patent No. 5,752,037).

Dependent claim 13 is rejected under 35 USC 103(a) as being unpatentable over Arimilli, Bourekas, Gornish, and further in view of Steely (US Patent No. 5,966,737).

Claims 1, 4, 7, 12, 14, 15 and 16 are amended.

Thus, claims 1-16 remain pending for reconsideration, which is respectfully requested.

No new matter has been added in this Amendment. The foregoing rejections are hereby traversed.

INDEPENDENT CLAIMS 1 AND 14

Claims 1-3 and 14 are rejected under 35 USC 103(a) as being unpatentable over Arimilli (US Patent No. 6,374,330) in view of Bourekas (US Patent No. 6,128,703). Arimilli and Bourekas are newly cited, and, thus, newly relied upon to reject independent claims 1 and 14.

Arimilli discloses three new "U" states, for upstream, undefined sector, as part of the MESI protocol to maintain cache coherency in a multi-processor system implementing a sectored cache between each processor and the system bus (i.e., a multi-level cache hierarchy). In particular, Arimilli's "U" states are directed to improving tracking sectors that are valid in higher cache levels without executing unnecessary bus operations, and more efficiently supporting cachable write-through operations. See, Arimilli, column 5, line 34 to column 6, line 65. The Examiner asserts that Arimilli discloses cache coherence of MESI protocol, but Arimilli does not disclose anything about the claimed present invention's "weak read operation for causing failure in said pre-fetch request as a pre-fetch protocol" as part of the MESI protocol. So the Examiner relies on Bourekas for the claimed present invention's "weak read operation for causing failure in said pre-fetch request as a pre-fetch protocol" as recited in independent claims 1 and 14.

Independent claims 1 and 14 are amended for clarity. In contrast to Arimilli and Bourekas, the claimed present invention provides, "a cache controller for carrying out, in ~~the~~a case that at ~~the~~a time of generation of a pre-fetch request following a read request from one of the processors the data stored in the other cache devices ~~cannot~~must be read ~~unless its state tag is changed~~by changing state tags of the other cache devices, weak read operation for causing failure in said pre-fetch request as a pre-fetch protocol" (amended claim 1). Support for the claim amendments, including dependent claims 2-3, can be found, for example, in paragraphs 7-10, of the substitute specification.

Bourekas discloses memory write-back of a cache line in the modified state (M) in the pre-fetch. More particularly, Bourekas discloses a processor that includes a primary cache and a prefetch instruction with an ignore-hit indicator. The ignore hit indicator provides an indication to the primary cache as to whether the specified data should be retrieved from the main memory, even if the specified data is stored within the primary cache. Bourekas discloses that the advantage of the ignore-hit indicator in the prefetch instruction is that in a multi-bus master environment, as shown in FIG. 3, which uses a microprocessor that does not have hardware snoop circuitry, to insure cache coherency, burst reads can be utilized without requiring the programmer to specify a separate operation to explicitly flush a cache before retrieving the prefetched data. See, column 5, lines 3-29, column 10, lines 14-22 and FIG. 8 (relied upon by the Examiner), and column 10, lines 35-46.

However, Bourekas cannot disclose or suggest the present claimed invention's "a cache controller for carrying out, ***in ~~the~~a case*** that at ~~the~~a time of generation of a pre-fetch request following a read request from one of the processors ***the data stored in the other cache devices ~~cannot~~must be read ~~unless its state tag is changed~~by changing state tags of the other cache devices, weak read operation for causing failure in said pre-fetch request as a pre-fetch protocol***," because Bourekas expressly discloses that its data coherency method applies to a multi-bus master environment without snoop circuitry, such that Bourekas does not cover a cache of multi-processors, but Bourekas covers a pre-fetch of a cache of a single processor, which differs from the claimed present invention as recited in independent claims 1 and 14. More particularly, Bourekas does not disclose or suggest cache coherence for a plurality of cache devices. In contrast to Bourekas, the claimed present invention provides, "cache device set up in ***each of processors***" and in which "a cache memory wherein a part of data in the main memory is stored in ***one or more cache lines and a state tag using to***

manage data consistency is set up in each of the cache lines." In other words, Bourekas' cache data coherency technique is not directed to the claimed present invention's cache data coherency technique in a multi-processor system.

According to independent claims 1 and 14, as amended for clarity, whether a read by a pre-fetch is successful depends upon ***what are the states of the other caches***. In Bourekas, on the other hand, cache invalidation depends upon whether the state of ***own*** cache is M. In other words, the claimed present invention fails a pre-fetch read when, "***the data stored in the other cache devices ~~cannot~~must be read ~~unless its state tag is changed~~by changing state tags of the other cache devices***" (amended independent claims 1 and 14). The subject matter of the claimed present invention is based upon a concept of whether the pre-fetch is successful, and if the pre-fetch disturbs the other cache devices, the pre-fetch is caused to fail.

More particularly, the claimed present invention as recited in independent claims 1 and 14, is characterized in that a cache controller carries out a "***weak read operation for causing failure in the pre-fetch request as a pre-fetch protocol***," if at a time of generation of the pre-fetch request following a read request from one of the processors, "***the data stored in the other cache devices must be read by changing state tags of the other cache devices***" (i.e., as recited in dependent claim 2, "weak read operation for causing failure in the pre-fetch request" is performed "when the data corresponding to the pre-fetch request stored in the other cache devices is in the data-modified state (M) or the exclusive state (E)"). See, paragraph 9 of the Substitute Specification. In contrast, dependent claim 3 recites, "reads, when the data corresponding to the pre-fetch request and stored in the other cache devices is in the invalid state (I), the same data from said main memory and stores the same data in the exclusive state (E) in the cache memory; and when the data is in the data-shared state (S), the cache controller reads the data from the other cache devices and stores the data in the data-shared state (S) in the cache memory," where the protocol in this case is based on the MESI protocol that is normal, as described in paragraph 10 of the Substitute Specification. Therefore, the claimed present invention's pre-fetch protocol achieving a "***weak read operation for causing failure in the pre-fetch request as a pre-fetch protocol***" has a benefit of causing exclusion of any pre-fetch resulting in a change in the states of the other cache devices, so as to reduce overhead accompanying the pre-fetch read request and writing on the data corresponding thereto.

In contrast to Bourekas, either alone or as combined with Arimilli, the claimed present invention is directed to a cache data coherence technique in a multiprocessor system as recited in independent claims 1 and 14, using claim 1 as an example:

> 1. (ORIGINAL) A cache device set up in each of processors, interconnected to other cache devices in other processors and connected to a main memory, which comprises:
>
> a cache memory wherein a part of data in the main memory is stored in ***one or more cache lines and a state tag using to manage data consistency is set up in each of the cache lines***, and
>
> a cache controller for carrying out, ***in ~~the~~<u>a</u> case*** that at ~~the~~<u>a</u> time of generation of a pre-fetch request following a read request from one of the processors ***the data stored in the other cache devices ~~cannot~~<u>must</u> be read ~~unless its state tag is changed~~<u>by changing state tags of the other cache devices,</u> weak read operation for causing failure in said pre-fetch request as a <u>pre-</u>fetch protocol*** (emphasis added).

Support for independent claims 1 and 14 can be found, for example, in paragraph 56, pages 31-33 of the substitute specification, and FIG. 10. Again, one benefit of the present claimed invention is that if state tag of a cache line must be changed to properly complete a prefetch request (i.e., maintain cache data coherence), then the prefetch request is terminated to reduce such overhead.

<u>INDEPENDENT CLAIMS 4, AND 15</u>

Independent claims 4 and 15 are rejected under 35 USC 112, first paragraph, for failing to comply with the enablement requirement. Further, dependent claims 5 and 6, which depend from claim 1 and recite similar features of independent claim 4, are rejected under 35 USC 103(a) as being unpatentable over Arimilli, Bourekas, and Prudvi. The forgoing 35 USC 112, first paragraph, and 35 USC 103 rejections are hereby traversed as follows.

Regarding the 35 USC 112, first paragraph, rejection, independent claims 4 and 15 are amended for clarity, by reciting:

> a cache controller for carrying out a pre-fetch protocol that in ~~the~~a case that at ~~the~~a time of generation of a pre-fetch request following a read request from one of the processors the data stored in the other cache devices ~~cannot~~must be read ~~without changing its state tag~~by changing state tags of the other cache devices, the data is read without changing the state tag and stored in the cache memory with ~~the~~ setup of a weak state W, and at ~~the~~a time of synchronization operation of memory consistency to attain data-consistency by software the data in the cache memory in said weak state (W) is wholly invalidated.

The Examiner in page 3 of the Office Action asserts: "it is not understood that if data stored in the cache devices cannot be read without changing its state tag, then why the data is read without changing the state tag?"

Independent claims 4 and 15 are supported by the second weak protocol processing unit 46 of the present invention, as described, for example, in paragraphs 12-15, paragraph 53 (page 26, lines 1-9), paragraph 57-59, of the Substitute Specification, and FIG. 12, operations 4 and 5.

In particular, the present claimed invention as recited in independent claims 4 and 15 provides a new "weak state W" state in a cache-coherency protocol.

In particular, for example, paragraph 59 (page 40, lines 1-10) of the Substitute Specification, expressly describes that when data in the exclusive (E) or the data modified state (M) are pre-fetched from the main memory to other cache devices (i.e., operation 5 in FIG. 12), it is also possible in the pre-fetch processing based on the second weak protocol read to ***prevent change in the states of the other cache devices from being caused***. In other words, the pre-fetched data can be forcibly invalidated regardless of the change in the states of the other cache devices by the synchronization operation in the memory consistency model. In this manner, it is possible to ***reduce***, without invalidating data newly, ***overhead at the time of writing data*** (i.e., overhead when, in the cache devices in which data in the exclusive state (E) or the data modified state (M) is stored, the writing of their processors is caused) (page 40, lines 1-10 of the Substitute Specification).

Therefore, a benefit of the claimed present invention is to provide a new "weak state W" in a cache-coherency protocol and treat the pre-fetch request as a success in a case that "the

data stored in the other cache devices ~~cannot~~must be read ~~without changing its state tag~~by changing state tags of the other cache devices," (i.e., when data in the exclusive (E) or the data modified state (M) are pre-fetched from the main memory to other cache devices), so that the order of memory transaction is renewed and results of writing are not reflected, thereby forcing a synchronization operation while preventing change in states of the other cache devices. In this manner, it is possible to ***reduce***, without invalidating data newly, ***overhead at the time of writing data*** (i.e., overhead when, in the cache devices in which data in the exclusive state (E) or the data modified state (M) is stored, the writing of their processors is caused). The present claimed invention's benefit of changing the order of memory transaction by regarding the pre-fetch request as a success in a case that data stored in the other cache devices cannot be read without changing its state tag, ***adopts a new type of weak consistency memory model according to the present invention***. See, paragraph 57 (page 34, lines 9-18), and paragraph 58 (page 37, lines 4-16), and paragraph 59 (page 40, lines 1-10), of the Substitute Specification).

Regarding rejection of independent claims 4 and 15 as being obvious over Arimilli, Bourekas and Prudvi, as also discussed above concerning the enablement rejection, according the claims 4 and 15, the cache controller carries out a pre-fetch protocol that in a case that at the time of generation of a pre-fetch request following a read request from one of the processor, the data stored in the other cache device must be read by changing the state tags of the other cache devices (for example, if the state tags in the other cache devices are M or E), the data is read without changing the state tags, and after responding to the processor, the data is stored in the cache memory with the setup of "a weak state W." Subsequently, at the time of synchronization operation of memory consistency, a pre-fetch protocol to wholly invalidate the data in the cache memory in the weak state (W) is executed.

The Examiner asserts that while Arimilli or Bourekas disclose nothing about setting of the claimed present invention's "weak state W," Prudvi's lazy state (L) corresponds to the claimed present invention's "weak state W." However, Prudvi treats the MESI protocol, which is shown in FIG. 3, as an SLIME protocol shown in FIG. 4. In Prudvi, when the read cache data is in the modified (M) state, it is changed into the lazy state (L). Prudvi's lazy state protects the data from exclusion from cache and represents the possibility to share with the other cache devices.

However, the claimed present invention as recited in independent claims 4 and 15 differs from Prudvi's SLIME cache coherency protocol in a multi-processor system, as follows.

Independent claims 4 and 15 recite an invention regarding the treatment of cache data at the time of "read of data" (i.e., read pre-fetch protocol), whereas Prudvi's SLIME relates to the treatment of "written data." Therefore, independent claims 4 and 15 and Prudvi are completely independent of each other, and it is even possible to use them together as different components for cache coherency in a multiprocessor system.

For example, the difference between the claimed present invention's W state and Prudvi's L state is apparent from the MESI state transition. The claimed present invention's W state, being a state for reading, is present only in the form of transition from the I state. However, Prudvi's L state, being a state for the written data, takes the form of transition from the M state. Of course, the claimed present invention and Prudvi are quite different in that Prudvi discloses nothing about the claimed present invention's invalidation upon synchronization (i.e., Prudvi does not disclose or suggest the claimed present invention's, "at ~~the~~a time of synchronization operation of memory consistency to attain data-consistency by software the data in the cache memory in said weak state (W) is wholly invalidated"). The W-state of independent claims 4 and 15 can be achieved by combining a weak consistency model, typically represented by a weak consistency, and a pre-fetch (paragraphs 57-58 of the Substitute Specification). In contrast, Prudvi's L state takes into account only the conventional strong consistency model (known as sequential consistency or strong consistency).

In other words, the present claimed invention's "weak state (W)" treat the pre-fetch request as a success in a case that data stored in the other cache devices cannot be read without changing its state tag, so that the order of memory transaction is renewed and results of writing are not reflected, thereby forcing a synchronization operation while preventing change in states of the other cache devices. Prudvi's Lazy Update state differs from the present claimed invention, because Prudvi's Lazy Update state indicates that the copy is protected from eviction in the cache and may be shared with other agents (column 3, lines 45-49). Further, Prudvi's Lazy Update state is not directed to the present claimed invention's cache data coherency concerning a read "pre-fetch protocol."

Therefore, independent claims 4 and 15 are not obvious over Arimilli, Bourekas and Prudvi, and dependent claims 5 and 6 are not obvious accordingly.

INDEPENDENT CLAIMS 7 AND 16

Claims 7-10 and 16 are maintained as rejected under 35 USC 112, first paragraph, for failing to comply with the enablement requirement. Independent claims 7 and 16 are amended for clarity.

Independent claims 7 and 16 are directed to the passive preservation mode embodiment of the present invention. The Examiner in page 3 of the Office Action asserts: "It is not understood that if the other cache devices do not contain the data corresponding to the pre-fetch data, then why the pre-fetch data get invalidated?"

Independent claims 7 and 16 are supported by the negative reading mode processing unit 48 of the present invention, as described, for example, in paragraphs 60-65, of the Substitute Specification, and FIGS. 7, 13-14. In particular, the present claimed invention as recited in independent claims 7 and 16 provides a new "passive preservation mode P" state in a cache-coherency protocol in a multi-processor system.

Paragraph 60 (page 40, lines 14-16) of the Substitute Specification, expressly discloses that "this passive preservation mode is a protocol for invaliding pre-fetched data when data in the address equal to the address of the pre-fetched data preserved in the cache is requested by some other cache device," which has a benefit of reducing useless sharing of read pre-fetched data by other cache devices, as described in paragraph 17 and paragraph 65 (page 45, lines 3-5) of the substitute specification. In other words, pre-fetch data is treated as speculative data, which is not shared unless a HIT line is asserted, and if a HIT line is not asserted, the pre-fetched data is invalidated (paragraph 20 and paragraph 64 (page 43, line 6 to page 44, line 6) of the Substitute Specification).

More particularly, the following description explains operation and benefit of the claimed present invention as recited in independent claims 7 and 16. Usually, when data is read in a state in which the other cache devices store data, the state becomes an S state. After the S state, data writing requires an invalidation operation of the other cache devices, requiring a considerable overhead. However, if none of the other cache devices hold data, the data can be stored in the E state. Data writing in the E state does not require an invalidation operation. Assume, for example, a case where the other cache devices are exclusively those holding data in the claimed present invention's P state. In this case, the cache device from which data is read is in the S state, thus requiring an invalidation operation upon writing. However, the P state

originally represents data read out for a pre-fetch of which the existence has no practical importance. Therefore only for this P state data, it would be useless to be forced to select the S state rather than the E state (i.e., the E state should be selected). In other words, when comparing the overhead for an invalidation request upon selection of the S state with the demerits of throwing away the pre-fetch data, the latter provides favorable merits. Independent claims 7 and 16, therefore, set forth a concept in which, if the "other cache devices" holding data, store only data in the P state, the data in the P state are thrown away, and the data are held in the E state in the cache devices which carry out read from the main memory.

Therefore, the claimed present invention as recited in independent claims 7 and 16 are enabling, and further, in contrast to the relied upon references, the present invention using claim 7 as an example provides:

> 7. (CURRENTLY AMENDED) A cache device set up in each of processors, interconnected to other cache devices in other processors and connected to a main memory, which comprises:
>
> ... a cache controller for carrying out a pre-fetch protocol according to a process comprising:
>
> ~~that~~setting as the state tag, at the time of generation of a pre-fetch request following a read request from one of the processors, a passive preservation mode P ~~is set up~~ to data pre-fetched from the other cache devices or the main memory, ~~and then~~
>
> storing the pre-fetched data ~~is stored~~ in said cache memory, [[;]]
>
> ***not informing,*** ***when the data corresponding to ~~the~~a read request from the other cache devices is the pre-fetch data to which said passive preservation mode P is set*** ~~up~~, the other cache devices ~~are not informed~~ of the preservation of the corresponding ~~data;~~data, and
>
> ***invaliding*** the pre-fetched data in the cache memory, ***when none of the other cache devices store the corresponding data, and preserving said pre-fetch data as it is,*** ~~***invalidated; and***~~ ***when the other cache devices share the corresponding data***~~, said pre-fetch data is stored as it is~~ (amended independent claim 7, emphasis added, FIGS. 13 and 14, operations 3 and 4).

DEPENDENT CLAIM 12

Dependent claim 12 is rejected under 35 USC 112, second paragraph, for indefiniteness. Dependent claim 12 is amended, taking into consideration the Examiner's comments. Support for dependent claim 12 can be found, for example, in FIG. 15 and paragraph 66 of the Substitute Specification. Withdrawal of the indefiniteness rejection of claim 12 is respectfully requested.

CONCLUSION

In view of the claim amendments and the remarks, withdrawal of the rejection of pending claims and allowance of pending claims is respectfully requested.

There being no further outstanding objections or rejections, it is submitted that the application is in condition for allowance. An early action to that effect is courteously solicited.

Finally, if there are any formal matters remaining after this response, the Examiner is requested to telephone the undersigned to attend to these matters.

Respectfully submitted,
STAAS & HALSEY LLP

Date: September 13, 2004

By: [signature]
Mehdi D. Sheikerz
Registration No. 41,307

1201 New York Avenue, NW, Suite 700
Washington, D.C. 20005
Telephone: (202) 434-1500
Facsimile: (202) 434-1501